User's Manual for US1 Module of the NUFT Code, Version 1.0

(Single-phase, Isothermal Unsaturated Model)

John J. Nitao
Earth Sciences Department
Lawrence Livermore National Laboratory

date: March 20, 1996

time: 3:30 PM

file: us1.tex

Contents

1 Introduction . . . 3

2 How to Install the Model . . . 5

3 How to Run the Model . . . 7

4 Input Format . . . 8

5 Input Data for `us1p` Flow Model . . . 10

`time, tstop, dt, dtmax, stepmax` – *Time step control* . . . 11

`tolerdt, reltolerdt` – *Time step control* . . . 12

`tolerconv, reltolerconv` – *Newton-Raphson convergence tolerances* . . . 13

`genmsh` – *Generate mesh* . . . 14

`state` – *Initial Conditions* . . . 17

`restart` – *Read and write restart information* . . . 18

`rocktab` – *Material properties* . . . 20

`srctab` – *Sources and sinks* . . . 23

`bctab` – *Boundary conditions* . . . 24

`output` – *Output options* . . . 25

Other options . . . 30

5.1 Notes: Table Time Values . . . 31

6 Input Data for `us1c` Transport Model . . . 32

`state` – *Initial conditions* . . . 32

`rocktab` – *Material properties* . . . 33

`srctab` – *Sources and sinks* . . . 34

`bctab` – *Boundary conditions* . . . 35

`compprop, phaseprop` – *Component and phase properties* . . . 36

`decay` – *Contaminant decay* . . . 37

7 References . . . 38

A The Mathematical Model . . . 39

A.1 Variable Switching . . . 40

A.2 Numerical Discretization . . . 40

B Sample Problems . . . 41

B.1 Flow Problem . . . 41

B.2 Transport Problem 44

1. Introduction

This manual describes the US1 module of NUFT, which is a model for solving the flow and transport equations under isothermal conditions for unsaturated flow of a liquid phase. The flow of the gas phase is neglected and is assumed to be at constant pressure. The resulting governing equation often called Richard's equation. The transport of a single dilute non-volatile contaminant species in the aqueous phase can be optionally modeled. The liquid density is assumed to be approximately a constant (incompressible fluid) with regards to gradients of mass fractions with regards to calculating molecular diffusive fluxes. It is also assumed that the concentration of the contaminants are so low that the density of the liquid phase is essentially unaffected by the concentration of the contaminant. Because of this assumption, the flow and transport portions of the calculation in the code can be decoupled and performed sequentially.

NUFT (Nonisothermal Unsaturated-Saturated Flow and Transport model) is a suite of multiphase, multicomponent models for numerical solution of non-isothermal flow and transport in porous media with application to subsurface contaminant transport problems. These distinct models are imbedded in a single code in order to utilize a common set of utility routines and input file format.

Currently, the code runs on the Unix and DOS operating systems. Versions have been successfully compiled and tested for IBM-PC compatibles, Cray Unicos, and the following workstations: Sun, Hewlett-Packward, IBM Risc/6000, Silicon Graphics. Each set of related models is called a module and has its own user's manual which documents any particular features and input data specific to that module. The NUFT Reference Manual [Nitao, 1993] documents the general numerical algorithms used and gives the documentation of the input to the model common to all or most modules including options not described in the user's manual for each module.

Available modules are:

- `UCSAT` – unconfined and confined saturated flow model
- `US1P` – single phase unsaturated flow (Richard's equation)
- `US1C` – single component contaminant transport
- `USNT` – NP-phase, NC-component with thermal option.

An integrated finite-difference spatial discretization is used to solve the balance equations. The resulting non-linear equation is solved at each time by the Newton-Raphson method. Options for solution of the linear equations at each iteration are direct banded solution and preconditioned conjugate gradient method with various preconditioning schemes.

The model can solve one- , two- , or three-dimensional problems. Future plans include incorporation of capillary hysteresis, non-orthogonal mesh discretization, finite elements, non-linear solid sorption isotherms, and chemical reactions.

The first stage of code verification with one-dimensional problems has been completed (Lee et al., [1993]) and further verification efforts are planned.

The distinct models in the code utilize a common set of utility routines and input file format. The various models are essentially isolated from each other and hence future models can be added without affecting existing models. This also allows for easy maintenance and future enhancements. Global variables in the code are virtually non-existent. The code is written principally in the C language. Input data is in the form of that used by the lisp language. An internal lisp interpreter for the Scheme dialect of lisp is part of the simulator whose purpose is

to read the input data file and the internal data files containing default input data values. It also performs data checking.

This manual is self-contained and describes a minimal set of the most commonly used input parameters necessary for the use of this module. The NUFT reference manual (Nitao, [1995]) contains generic input options common to all of most of the modules. It also contains further input options not given in this manual.

acknowledegments

The author wishes to thank the management of the Environmental Restoration Division at the Lawrence Livermore National Laboratory (LLNL) for supporting the documentation and verification of the NUFT code. The basic concepts of the code was developed under the funding of the LLNL Institutional Research and Development program.

2. How to Install the Model

Installation onto a Unix System
The distribution of NUFT for the Unix operating system (Cray Unicos and unix workstations) comes with a C-shell installation script called INSTALL-SCRIPT and a compressed tar file. The name of the compressed tar file is of the form `xxx.tar.Z` where `xxx` are characters referring to the NUFT version number. Copy these two files to any directory (for example, your home directory).

1. Type:

   ```
   csh INSTALL-SCRIPT tar-file
   ```

2. The directory `bin` just below your home directory must be placed as part of your execution path by editing the PATH variable that is usually set in the `.login` file in your home directory, if it is not already there. For example,

   ```
   PATH = .:$HOME/bin:/bin:/usr/bin:/ucb/bin
   ```

3. The installation script and the tar file can be deleted after successful installation. See below for the location of the tar file after installation.

The installation script will create directories `bin` and `NUFT` below your home directory unless they exist already. The installation script creates a directory with the name `xxx` below the NUFT directory derived from the name of the compressed tar file `xxx.tar.Z`. It then copies the compressed tar file and extracts its contents into this directory. It then symbolically links the file `nuft` to the file `nuft-dist` in the installed directory. This is the file that your system executes when you type the `nuft` command which runs the model. NUFT sample input files are also placed into the installed directory.

The script does not remove the directories containing older NUFT versions. The user may delete older versions if they wish. The installation script will give a message indicating whether the installation was successful. After successful installation, the model is ready to run. See the next section in order to run the model.

Installation for IBM-PC Compatibles under DOS
The NUFT distribution for IBM-PC Compatibles under DOS comes on a floppy disk with an installation script called `install.bat`.

1. Type:

   ```
   a:\install.bat
   ```

   with the floppy in the `a:` drive.

2. The directory `\nuft` must be placed as part of your execution path by editing the PATH variable that is set in the `autoexec.bat` file in the top directory, if it is not already there. For example,

   ```
   PATH=C:.;C:\;C:\DOS;\nuft
   ```

The installation script will create directories `\nuft` below your home directory unless it exists already. The installation script then creates a directory with the name `xxx` equal to the NUFT

version number and copies the appropriate files there. A file called `nuft.bat` is copied to the `\nuft` directory. This is the file that your system executes when you type the `nuft` command which runs the model. NUFT sample input files are also placed into the installed directory.

The script does not remove the directories containing older NUFT versions. The user may delete older versions if they wish. The installation script will give a message indicating whether the installation was successful. After successful installation, the model is ready to run. See the next section in order to run the model.

3. How to Run the Model

Steps to Run the Model:

1. **Install the code.** See the previous section on how to install the code.
2. **Specify the mesh.** The mesh can be created through either of the following methods.
   - Using the `genmsh` option in the input file
   - Using an external program that creates a mesh file (see the NUFT reference manual for description of the mesh file format).

   The first option is recommended for new users.
3. **Create the input data file.** The input file is created using any ascii text editor. An editor such as `vi` or `emacs` which signals matching parentheses is preferable. You will need to read section on the input data syntax to understand the general format of the types of input data that goes into the input file. For first time users it is easiest to edit the preexisting sample input files provided with the code distribution.
4. **Run the model.** Type:

   `nuft` *input-file*

4. Input Format

Before going further the user should read the sections on the input data syntax in the general users manual for NUFT. The model can be used either to solve for flow of the liquid phase only or for simultaneous flow including transport of a contaminant.

Form of Input Data for Flow Only:

```
(us1p
     (title ...)       ;; run title
     (output-prefix ...)  ;; prefix of all output files
     (time ...)        ;; initial time
     (tstop ... )      ;; ending time
     (dt ...)          ;; initial time step size
     (dtmax ...)       ;; maximum time step
     (stepmax ...)     ;; maximum no. of time steps
     (genmsh ...)      ;; mesh specification
     (state ...)       ;; initial conditions
     (rocktab ...)     ;; soil property type
     (output...)       ;; output specification
     (srctab ...)      ;; source term tables
     (bctab ...)       ;; boundary condition tables
)
```

The '...' denote data which will be explained later. All input past a semicolon on a given line is treated as comments. The above data units can occur in any order except for the word `us1p` which must come first. Either `state` or `read-restart` must be present, but not both. The data items `(output ...)`, `(srctab ...)`, and `(bctab ...)` are all optional; the rest are required. There are other optional data items not shown above but are described in the NUFT Reference Manual (Nitao, [1995]).

Flow and transport is solved sequentially. That is, at each major time cycle, the code first solves for the flow of the phase and then solves for the transport of the contaminant in an alternating fashion. (The model has the capability to allow the transport calculation to take different time step sizes than the flow so that several time steps may be taken by the transport model in a single major time cycle, if needed.)

Form of Input Data for Flow and Contaminant Transport:

```
(common                          ;; data common to flow and transport
     (title ...)
     (output-prefix ...)
     (output-ext ...)
     (genmsh ...)
     (time ...)
     (tstop ...)
 )

(us1p                            ;; flow input data
     (dt ...)
     (dtmax ...)
     (stepmax ...)
     (state ...)
     (rocktab ...)
```

```
        (output...)
        (srctab ...)
        (bctab ...)
    )

    (us1c                              ;; transport input data
        (dt ...)
        (dtmax ...)
        (stepmax ...)
        (state ...)
        (rocktab ...)
        (compprop ...)
        (output...)
        (srctab ...)
        (bctab ...)
    )
```

The first principal data unit `common` holds input data that is common to both the the flow and transport models including the mesh generation data. The second unit `us1p` holds the input fo the flow model and, except for data now in the the `common` unit, is the same as for the flow-only mode. The third unit `us1c` holds the input parameters for the transport model. The two data units `phaseprop` and `compprop` are will be described in a later section of this manual.

5. Input Data for us1p Flow Model

In this section we describe the input data needed to run the us1p model. Some of the data requirements are shared by other models. Not all of the options are given here for such types of data but are documented in the *NUFT Reference Manual.*

Time step control.

`(time` *<start-time>*`)`

<start-time> *t-real*

initial starting time of run. For restarts, this time is not needed if `restart` is set, in which case the initial time will be set to the time read in through the restart file. If `time` appears, this overwrites the time read in.

`(tstop` *<stop-time>*`)`

<stop-time> *t-real*

run will stop when it reaches this time.

`(dt` *<initial-time-step>*`)`

<initial-time-step> *t-real*

initial time step size.

`(dtmax` *<max-step-size>*`)`

<max-step-size> *t-real*

maximum time step size.

`(stepmax` *<max-steps>*`)`

<max-steps> *integer*

maximum number of time steps; run will stop if this will be exceeded.

Time step control.

(tolerdt
<*tolerdt-S*>
<*tolerdt-H*>
)

Set absolute tolerances for time step control.

<*tolerdt-S*> *real*
<*tolerdt-H*> *real*

The model will control time step such that the magnitude of changes in the primary variables: S (fraction) and head H (meters) from one time step to the next so they do not exceed the specified values. As a rule of thumb the tolerances should be about one to two orders of magnitude larger than the desired accuracy of the primary variable. Default tolerance for pressure is 10 m. pressure and 0.35 for liquid saturation.

(reltolerdt
<*reltolerdt-S*> <*reltolerdt-H*>

Set relative tolerances for time step control. The model calculates a time step based on the absolute tolerance set in tolerdt and another time step based on relative tolerances set in reltolderdt and uses the larger of the two.

<*reltolerdt-S*> *real*
<*reltolerdt-H*> *real*

The model will also control time step such that the relative magnitude of changes in the primary variables, liquid saturation S and head H, from to the next does not exceed the specified values. For each primary variable the model calculates the largest magnitude over all the elements in the entire problem domain and then multiplies by the specified relative tolerance to get a maximum change. Default tolerance for head is 0.1 and 0.0 for saturation.

Newton-Raphson convergence tolerances.

(tolerconv
<tolerconv-S> <tolerconv-H>
)

(*optional*) Sets tolerance for Newton-Raphson iteration convergence criteria. The model checks for convergence using both the tolerdt and reltolderdt parameters. Convergence is assumed if either one of the criteria is true for each primary variable.

<tolerconv-S> *real*
<tolerconv-H> *real*

Absolute criteria for convergence of the Newton-Raphson iterations for solution of the non-linear, implicit-in-time, discretized balance equations is acheived if the magnitude of the changes in the primary variables, saturation S and head H (meters) from one iteration to the next is less than or equal to these values. The default tolerance is 0.1 m for the head and 0.001 for saturation.

(reltolerconv
<reltolerconv-S> <reltolerconv-H>
)

<reltolerconv-S> *real*
<reltolerconv-H> *real*

Criteria for Newton-Raphson convergence based on the maximum relative magnitude of changes in the respective primary variables. Default values are 0.001 for pressure and 0.0 for saturation.

Generate mesh.

```
(genmsh
     (coord  <mesh-type>)
     (down <Vx> <Vy> <Vz>)
     (dx <dx-1> <dx-2> ...)
     (dy <dy-1> <dy-2> ...)
     (dz <dz-1> <dz-2> ...)
     (mat
          (<el-name-prefix><mat-type>
               <i0><i1> <j0><j1> <k0><k1>)
                    ...
          (<el-name-prefix><mat-type>
               <i0><i1> <j0><j1> <k0><k>)
     )
     (anisotropic)
     (wrap-around)   optional
)
```

specifies the mesh geometry, element material types and names.

<mesh-type> *word*

specifies the type of mesh that will be generated. Valid options:

`rect`	rectangular mesh
`cylind`	cylindrical mesh

Let the three coordinates of our mesh system be x, y, and z. If the mesh type is `rect`, then x, y, and z are along the coordinate axes of a rectangular system. If mesh type is `cylind`, then the first coordinate x is the radial distance r, the second coordinate y is angle θ, and the third coodinate z is the longitudinal axis.

<Vx> <Vy> <Vz> *reals*

are the components of the vector pointing downward in the direction of the gravity vector. The program will internally normalize the vector to unity. Setting the components to all zero will turn off gravity in the model. The vector is always with respect to a rectangular coordinate system (X, Y, Z). For a rectangular mesh, the coordinate system coincides with the rectangular coordinate system (x, y, z) of the mesh. If the mesh is cylindrical, the vector is with respect to a coordinate system (X, Y, Z) where X is the axis defined by $\theta = 0, z = 0$; Y is the axis defined by $\theta = 90°, z = 0$, and the axis Z is defined by $r = 0$.

<dx-1> <dx-2> ... *reals*

are the subdivisions of the mesh in the direction of the first coordinate x. For a rectangular mesh, the subdivisions are in the increasing x direction of the rectangular coordinate system of the mesh. For a cylindrical mesh, the subdivisions are in the increasing r direction. Numbers that are repeated can be abbreviated; for example, `3*5.0` would stand for three repeats of the numeral `5`, that is, `5.0 5.0 5.0`.

<dy-1> <dy-2> ... *reals*

are the subdivisions of the mesh in the direction of the first coordinate x. For a rectangular mesh, the subdivisions are in the increasing y direction. For a cylindrical mesh, these parameters set the subdivisions in the increasing θ direction. Angular units are in *degrees*. The cylindrical mesh wraps completely around the θ direction only for the case of a single angular subdivision whose value is equal to *dy-1* = 360°).

<dz-1> <dz-2> ... *reals*

are the subdivisions of the mesh in the increasing z direction for either a rectangular or cylindrical mesh system.

<el-name-prefix> *word*

prefix of element name. Each element in the specified i, j, k index range will a name comprised of this prefix and a number associated with it as the suffix (see the explanation of index ranges below for examples). This prefix can be used to specify a range of elements in the `state`, `output`, and other input parameters. The element names will be of the form *<el-name-prefix>*`#`*<i>*`:`*<j>*`:`*<k>* where *<i>*,*<j>*,*<k>* denote the i, j, and k indices.

<mat-type> *word*

material type. Each element in the specified i, j, k range will have this material type (see the explanation of index ranges below for examples). Each material type must have a corresponding entry in the material type table in `rocktab`. An exception are material types that are pre-defined in the model. At this time, the only pre-defined material type is `NULL` which completely removes the respective element or elements from the model. An equivalent method of removing elements is the `null-blocks` option describedin the NUFT Reference Manual. Material types that are of the following form denote specific 'auxillary' elements used for boundary conditions,

`BC.1.xxx` element for b.c. of first type,
`BC.2.xxx` element for b.c. of second type,
`BC.3.xxx` element for b.c. of third type.

An element for first and third type boundary conditions will have zero flow connection from the element centroid to the boundary of any neighboring non-auxillary elements. An element for a second type boundary condition will have flow connection set to unity so that the reciprocal of the permeability or hydraulic conductivity will be the resistance, and the volume of the element is set to the average volume of neighboring non-auxillary elements.

<i0> <i1> <j0> <j1> <k0> <k1> *integers*

specifies a range of elements using the i, j, k indices of the element in the direction of the x, y, z directions. Used to specify the range of elements for setting element name prefixes *<el-name-prefix>* and the material types *<mat-type>*. **A specification record of the form,**

(*<el-name-prefix><mat-type><i0><i1><j0><j1><k0><k1>*)

will overwrite the effect of any previous records (see example below). In this way, one can specify non-rectangular regions to have the same element prefix or material type.
The i, j, k indices are consistent with the `dx`, `dy`, and `dz` parameters. That is,

the dx parameter specifies the x dimension of the elements for $(i = 1, 2, \ldots)$, in that order. Similarly, dy specifies the y dimensions of the elements for $(j = 1, 2, \ldots)$, and dz specifies the z dimensions for $(k = 1, 2, \ldots)$.
Example:

```
(e silt 2 4 3 10 1 8)
```

This example specifies that the elements in the index range (i = 2 *to* 4), (j = 3 *to* 10), and (k = 1 *to* 8) will have element names e1, e2, ... and will be of material type silt.
Example:
consider a mesh with 4 elements in the i direction, 3 elements in the j direction, and 5 elements in the k direction,

```
(mat
     (s  silt  1 4  1 3  1 5)   ;; first set all elements
                                ;; to silt
     (c  clay  1 4  1 3  2 2)   ;; make layer k=2 into clay
     (w  well  2 2  3 3  2 2)   ;; make i=2,j=2, k=2 into
                                ;; well element
     (aquif silt 1 4  1 3  5 5) ;; call lowest layer by
                                ;; different prefix
 )
```

The symbols nx, ny, and nz can be used anywhere in place of a number where an index is required. The model interprets these to mean the number of subdivisions in the x, y, and z directions, respectively.

(anisotropic)

this parameter is optional. (The older form of the parameter isot-dir can also be used instead.) Affects the choice of the permeability (or hydraulic conductivity) parameter. See the documentation of the parameters Kx, Ky, and Kz below. This parameter should not be present for models where isotropic permeability (or hydraulic conductivity) is desired unless the three permeability parameters for each material type are equal.

(wrap-around)

if present, model wrap the grid around in the angular direction. I.e. for each element at $j = 1$ connect to the corresponding adjacent element at $j = ny$ where ny is the number of elements in the j direction. Default is no wrap-around.

Initial Conditions.

```
(state
     (<state-var-name>  <method>  <data>)
        ...
     (<state-var-name>  <method>  <data>)
)
```

initialize state variables.
<state-var-name>

The valid primary, or state, variable names are

H	pressure head above ambient (meters)
Sl	liquid saturation (fraction)

Note that the datum of H is chosen such that ambient pressure is H equal to zero.
The pressure head H is not initialized for the elements that are unsaturated. Set H equal to zero for these elements. The saturation Sl is set the desired initial saturation for those elements. For elements that are saturated, set H to the pressure head and Sl to any value greater than or equal to the maximum saturation.
(The general rules for setting the initial values of H and Sl are as follows. If for an element, H is set to greater than zero, then the liquid saturation value is overridden and set to the maximum saturation for that element. Otherwise, if Sl is set greater than or equal to the maximum saturation or if H is set to a number less than zero, then Sl is set to the maximum saturation, and H is set to zero.)

Example:

```
(state
     (Sl by-ztable
            ;; height
            ;; above
            ;; WT     Sl
               (0.0   1.0
                0.2   0.9
                0.8   0.5
                1.0   0.4
               10.0   0.3
              )
     )
     (Sl by-key ("bc*" 1.0))
     (H by-key  ("*" 0.0))
     (H by-key  ("bc*" 0.5))
)
```

First, Sl is initialized by interpolating from a table specifying values of saturation vs. height above the water table. Then, saturation is for elements with names starting with the characters "bc" is set to unit. The head is first initialized to all zeros, and then the elements with names starting with "bc" are initialized to 0.5 meters so that they are saturated elements.

Read and write restart information.

```
(read-restart
     (file "<file-name>")
     (time <restart-time>)    optional
)
```

"<*file-name*>" string

name of the restart file created by the option (output ...(restart ...) ...).

<*restart-time*> *real*

Time used to search in the restart file; the first restart record with time ¿= <*restart-time*> will be read in. The initial time of the simulation run is set by time and overwrites the time read in through the restart file. If time is not present, the initial of the run is set to the time read in.

Overwriting Restart Data

```
(overwrite-restart
     ...
)
```

The overwrite-restart statement is used to overwrite restart information read in from the read-restart statement for selected elements or variables. The overwrite-restart options are identical to those for the state statement.

Restart Backup

The model will periodically write out restart records so that the model can be restart in case of system failure. The model alternately writes out to two restart files named <*prefix*>.re0 and <*prefix*>.re1 where <*prefix*> is set by output-prefix. Each file contains a single record; previous records are overwritten. The reason for using two files instead of a single one is to prevent losing a record if the system fails during a write. The user must check the two files to see which has the record with the latest simulation time. The model writes to a file at periodic intervals based on the wall clock time.

(backup <*option*>) *optional*

<*option*> *word*

if set to on, then the model will periodically write backup restarts. If set to off, the model will not do backups. Default is on.

(backup-period <*backup-time*>) *optional*

<*backup-time*> t-real

wall clock time period for model to perform backup restarts. Default value is 10m (i.e., 10 minutes).

Material properties.

```
(rocktab
     (<rock-type name>
          (porosity <phil>)
          (Kx <hycond-0>) (Ky <hycond-1>) (Kz <hycond-2>)
          (pc  <pc-func> <pc-parameters>)
          (kr  <kr-func> <kr-parameters>)
     )
     ...
     (<rock-type name>
          (porosity <phil>)
          (Kx <hycond-0>) (Ky <hycond-1>) (Kz <hycond-2>)
          (pc  <pc-func> <pc-parameters>)
          (kr  <kr-func> <kr-parameters>)
     )
)
```

set material properties for flow model.

<rock-type-name> *word*

the name of the material.

<phi> *real*

the fractional porosity of the material. Must greater than zero.

<hycond-0> *real*
<hycond-1> *real*
<hycond-2> *real*

specifies the hydraulic conductivity (m/s) of material property. The model normally will use only the value of *<hycond-0>*. The other two values are ignored although they must be present and can be set to zero. However, if the parameter `isot-dir` is present in data unit `genmsh`, then the value of *<hycond-0>* will be used for flow in the first coordinate direction (for example, x-axis direction for rectangular coordinates and r-axis for cylindrical), *<hycond-1>* will be used for the second coordinate direction (y or θ), and *<hycond-2>* for the third coordinate direction (z).

<pc-func> *word*
<pc-parameters> *list of data units*

name of the matric potential head vs. saturation function and its associated parameters (the matrix potential is the approximately the same as capillary pressure except for saturations near residual). Valid options for `(pc` *<pc-func>* *<pc-parameters>*`)` are:

`(pc pcVanGen (m` *<m>*`) (alpha` *<alpha>*`) (Sr` *<Sr>*`) (Sj` *<Sj>*`))`

for the van Genuchten formulation.

<m>

parameter (unitless).

<alpha>

parameter (1/meters).

<*Sr*>

residual saturation.

<*Sj*>

optional transition saturation below which a linear curve is used. Default is $Sj = Sr + 0.05 * (1 - Sr)$.

`(pc pcTable (table` <*S0*>`) (Sr` <*pc0*>`)` <*S1*>`) (Sr` <*pc1*>`) ...`<*SN*> <*pcN*>`))`

table look-up option using linear interpolation. If liquid saturation is less than first entry, `S0`, then a value equal to `pc0` is used. If the liquid saturation is greater than the last entry `SN`, the value is `pcN`. <*Si*>

table entries for liquid saturation. They must be in strictly increasing order.

<*kri*>

table entries for capillary pressure head (meters).

The matrix potential function can be set to a *constant function* equal to the value <*real*> by replacing the above form, `(pc` <*pc-func*> <*pc-parameters*>`)` by:

`(pc` <*real*>`)`

where <*real*> is in head (m).

<*kr-func*> *word*
<*kr-parameters*> *list of data units*

name of the relative conductivity function vs. saturation function and its associated parameters. Valid options for `(kr` <*kr-func*> <*kr-parameters*>`)` are:

`(kr krVanGen (m` <*m*>`) (Sr` <*Sr*>`))`

for the van Genuchten formulation.

<*m*>

parameter (unitless).

<*Sr*>

residual saturation.

`(kr krTable (table` <*S0*>`) (Sr` <*kr0*>`)` <*S1*>`) (Sr` <*kr1*>`) ...`<*SN*> <*krN*>`))`

table look-up option using linear interpolation. If liquid saturation is less than first entry, `S0`, then a value equal to `kr0` is used. If the liquid saturation is greater than the last entry `SN`, the value is `krN`. <*Si*>

table entries for liquid saturation. They must be in strictly increasing order.

<kri>

table entries for relative permeability.

The relative conductivity function can be set to a *constant function* equal to the value *<real>* by replacing the above form, `(kr` *<kr-func>* *<kr-parameters>*`)` by:

`(kr` *<real>*`)`

Sources and sinks.

```
(srctab
     (compflux  (comp water)
                (range "<elem-range>" "<elem-range>" ...)
                (table  <flux-table>)
     )
     ...
     (compflux  (comp water)
                (range "<elem-range>" "<elem-range>" ...)
                (table  <flux-table>)
     )
)
```

specifies the component mass flux into an element or range of elements through a table of source fluxes at specified points in time. Linear interpolation is used for time intervals between the table values. Positive flux is flux *into* an element; negative flux is *out* of an element. Flux is specified as volumetric flux (cu.m./s) at ambient conditions.

Boundary conditions.

```
(bctab
     (<bc-name>
          (basephase liquid)
          (range "<elem-range>" "<elem-range>" ...)
          (tables
               (Sl <var-table>)
               (H <var-table>)
          )
          (factor  (water <factor-table>))
     )
     ...
     (<bc-name>
          (basephase liquid)
          (range "<elem-range>" "<elem-range>" ...)
          (tables
               (Sl <var-table>)
               (H <var-table>)
          )
          (factor  (water <factor-table>))
     )
)
```

specifies boundary condition information for flow model.

Output options.

Valid output variables for the us1p module are:

`H`	liquid head (m)
`S`	liquid saturation
`mc`	volumetric moisture content
`Kx, Ky, Kz`	hydraulic conductivities (m/s) (see `rocktab` for definitions)
`log10Kx, log10Ky, log10Kz`	logarithm base 10 of hydraulic conductivities (log10(ms/s))
`porosity`	porosity

```
(output

     (field
          (file-ext "<out-ext>")
          (format <options>)
          (index-range (<i0> <i1> <j0> <j1> <k0> <k1>)
                       ...
                       (<i0> <i1> <j0> <j1> <k0> <k1>)
          )
          (variables <el-var0> <el-var1> ...)
          (outtimes <t0> <t1> ...)
     )

     (flux-field
          (file-ext "<out-ext>")
          (format <options>)
          (index-crange
                ( (<i0> <j0> <k0>  <i1> <j1> <k1>)
                  ...
                  (<i0> <j0> <k0>  <i1> <j1> <k1>)
                )
          (variables <con-var0> <con-var1> ...)
          (outtimes <t0> <t1>  ...)
     )

     (history
          (file-ext "<out-ext>")
          (variable <el-var>)
          (element "<element-name>")
     )

     (flux-history
          (file-ext "<out-ext>")
          (variable <con-var>)
          (index-con <i0> <j0> <k0> <i1> <j1> <k1>)
     )

     (srcflux
          (comp <comp>)
          (file-ext "<out-ext>")
          (outtimes <t0> <t1> ...)
          (cumulative)    optional
     )
```

```
(bcflux
     (comp <comp>)
     (file-ext "<out-ext>")
     (outtimes <t0> <t1> ...)
     (cumulative)    optional
)

(forcetimes
     (outtimes <t0> <t1> ...)
)

(restart
     (file-ext "<out-ext>")
     (outtimes <t0> <t1> ...)
)

(extool
     (file-ext "<out-ext>")
     (index-range (<i0> <i1> <j0> <j1> <k0> <k1>) ... )
     (variables <var0><var1> ...)
     (outtimes <t0> <t1> ...)
)
)
```

specifies what variables will be dumped to files and the output format. Any of the above output options (`field`, `flux-field`, etc.) are optional or can be present any number of times in any order including with regards to different variables. (See below regarding other options and important notes.) Output options are:

`field`	output variables of selected elements at specified times
`flux-field`	output fluxes between selected elements at specified times
`history`	output variables of selected elements at all times
`flux-history`	output the fluxes between selected elements at all times (not applicable to USNT model)
`srcflux`	output component fluxes for a flux specified in `srctab`
`bcflux`	output component fluxes flowing out of a set of boundary elements specified by `bctab`
`forcetimes`	force the model to hit the times specified in `outtimes`
`restart`	write out a restart record to a file that can be read in by `restart` option
`extool`	write out data in *extool* format.

<out-ext> *string*

suffix of output file (`file-ext` is optional). The prefix of the output file name will be made up of the prefix set by the `output-prefix` parameter. (The suffix should be some mnemonic, e.g. the name of the variable being outputted.) For example, if `output-prefix` is set to "runA"and `file-ext` to ".C"then output will go to file "runA.C".
If the `file-ext` parameter is not present, then the output will be placed into the main output file. Output for different output options can share the same output file.

<options> *word*

format of output. Can be any of following:

`list` or `by-list`	output values at elements along with the element names (contsac format
`by-set`	list element values as a vector of the form [#n v1 v2 ...vn]

`by-x`	list x coord. and value
`by-y`	list y coord. and value
`by-z`	list z coord. and value
`by-ijk`	list i,j,k index and value
`by-xtable`	format compatible with `state` using by-xtable method, user needs to comment out output header
`by-ytable`	format compatible with `state` using by-ytable method, user needs to comment out output header
`by-ztable`	format compatible with `state` using by-ztable method, user needs to comment out output header
`tabular`	multi-column format
`contour`	format readable by the `nview` program for MS-DOS

<index-range> list of *integers*

range of elements specified by the i, j, k indices.

<comp> *word*

component name.

<el-var> *word*
<el-var0>, *<el-var1>* list of *words*

list of element variables that will be outputted (model specific).

<con-var> *word*
<con-var0>, *<el-var1>* list of *words*

list of connection variables (e.g. fluxes) that will be outputted (model specific).

<outtimes> list of *t-reals*

list of times at which output will be performed. Model will reduce the time steps to meet these times.

<index-crange> list of lists of *integers*

specifies a list of connections. Comprised of a list of list of integers with each list specifying a connection. Each list of integers is of the form

(*<i0> <j0> <k0> <i1> <j1> <k1>*)

and denotes the connection between element i =*<i0>*, j =*<j0>*, k =*<k0>* and element i =*<i1>*, j =*<j1>*, k =*<k1>*. Positive flux denotes flux from the first element to the second.

<out-var-name> *word*

name of output variable to be outputted by `history` option (model specific).

"*<element-name>*" *string*

name of element whose state, or primary, variable will be outputted.

<index-con> list of *integers*

specifies the single connection between the element (i =*<i0>*, j =*<j0>*, k =*<k0>*) and the element (i =*<i1>*, j =*<j1>*, k =*<k1>*). Positive flux denotes flux flowing from the first element to the second.

<bc-name> *word*

name of boundary condition used in `bctab`.

<src-name> *word*

name of source term used in `srctab`.

`(cumulative)`

optional. If present, the cumulative flux is outputted. If not present, the instantaneous flux is outputted. Note that cumulative fluxes are reset to zero at the beginning of a restart.

Notes:

1. Instead of `(file-ext "`*<out-ext>*`")` one can use `(file "`*<file-name>*`")` in order to explicitly specify the output file.
2. `(outtimes` *<t0> <t1> <t2>* ...`)` can be replaced by either of:

   a. `(outtimes *)` which means all times.

   b.

```
(triggers
    ((wake <state-var> (range "<el-name>"  "<el-name>" ...)  <op> <val>)
     (cond <state-var> (range "<el-name>"  "<el-name>" ...)  <op> <val>)
    )
    ...
    ((wake <state-var> (range "<el-name>"  "<el-name>" ...)  <op> <val>)
     (cond <state-var> (range "<el-name>"  "<el-name>" ...)  <op> <val>)
     )
)
```

   which checks to see at every time step if any of the triggers goes off. A trigger goes off iff its wake condition *<value> <op> <val>* is true where *<value>* is the value of the state variable *<state-var>* at the elements with names in the range of any of the pattern strings "*<el-name>*" .... If the wake condition is true, then the condition in the `cond` field is checked; if true then the trigger goes off and output occurs. Triggers can go off only once.

3. One can use `(range "`*<element-range>*"...`)` instead of `index-range` to specify elements by the names where "*<element-range>*" is a *pattern string*.
4. Similarly, one can use `(crange ("`*<element-range-0>*`" "`*<element-range-2>*`") ...)` instead of `(index-crange ((`*<i0> <j0> <k0> <i1> <j1> <k1>*`) ...))` to specify ranges of elements. Here, `("`*<element-range-0>*`" "`*<element-range-1>*`")` specifies all possible connections between elements whose names match the pattern string "*<element-range-0>*" and the elements whose names match "*<element-range-1>*".

5. One can use `(connection "`*<element-name-0>*`" "`*<element-name-1>*`")` instead of `(index-con` *<i0> <j0> <k0> <i1> <j1> <k1>*`)` to specify a connection.

Other options.

Placing the statement `(sat-lower-bound on)` into the input data unit for the flow model will tell the model to restrict fluxes coming out of an element, if necessary, in order that the saturation will stay above a certain value. The minimum value of saturation, <*slow-liq*>, is set by placing `(SlowbLiq` <*slow-liq*>`)` into each material type record in `rocktab`, i.e.,

```
(rocktab
     (<rock-type name>
          (SlowbLiq <slow-liq>)
          ...
     )
     ...
)
```

5.1. Notes: Table Time Values

Currently, the model may in some cases choose time steps which are so large that it overshoots sharp changes in a time table. This may be a serious problem if, for example, we want to model a flux that is turned off suddenly. A solution is to use the `forcetimes` command in `output` which forces the model to hit specified times, in this case, the times at which the boundary condition changes suddenly.

The last time value in table must be greater than or equal to the ending time of the simulation as set by `tstop` or the model will abort.

6. Input Data for `us1c` Transport Model

Initial conditions.

```
(state
     (<state-var-name>  <method>  <data>)
        ...
     (<state-var-name>  <method>  <data>)
)
```

<state-var-name>

The valid state, or primary variable, name is `C`, which is concentration. The concentration is in units of kg/m^3 (mass of contaminant/unit volume of phase) if `us1p` is used for providing the transport velocities and kg/kg (mass of contaminant/unit mass of liquid phase) if `usnt` is used.

Material properties.

```
(rocktab
     (<rock-type name> (porosity <phi>) (Kd <Kd>))
               ...
     (<rock-type name> (porosity <phi>) (Kd <Kd>))
)
```

set material properties for transport model.

<rock-type-name> *word*

the name of the material.

<phi> *real*

the fractional porosity of the material. Should be same value as set for flow model.

<Kd> *real*

dimensionless solid sorption coefficient defined as the quantity $\rho_B k_d/\phi$ where ρ_B is bulk dry density (g/cu.cm), k_d is the standard solid sorption coefficient (ml/g), and ϕ is porosity.

Sources and sinks.

```
(srctab
     (compflux  (comp <comp-name>)
                (name <flux-name>)
                (range "<elem-range>" "<elem-range>" ...)
                (table  <comp-flux-table>)
     )
     ...
     (compflux  (comp <comp-name>)
                (name <flux-name>)
                (range "<elem-range>" "<elem-range>" ...)
                (table  <comp-flux-table>)
     )

     (phaseflux (phase <phase-name>)
                (name <flux-name>)
                (range "<elem-range>" "<elem-range>" ...)
                (setcomp-internal)
                (table  <phase-flux-table>)
     )
     ...
     (phaseflux (phase <phase-name>)
                (name <flux-name>)
                (range "<elem-range>" "<elem-range>" ...)
                (setcomp-internal)
                (table  <phase-flux-table>)
     )
)
```

specifies the flux for component <*comp-name*> into an element or range of elements through a table of source fluxes at specified points in time. Linear interpolation is used for the flux at time intervals between the table values. The `compflux` option specifies the total mass flux of the component while `phaseflux` specifies the total mass flux of the phase. Positive flux is flux *into* an element; negative flux is *out* of an element. Units of component fluxes set in `comp-flux-table` are in kg/s while phase fluxes set in `phase-flux-table` are in units of cu.m/s and kg/s, respectively, for `us1p` and `usnt` providing the transport velocities. The `setcomp-internal` command specifies that the component fluxes in `phaseflux` are equal to the concentration of the respective element times the phase flux. Alternatively, a table of values can also be specified by the command:
`(setcomp (`<*comp-name*> `(table` <*t0*> <*C0*> <*t1*> <*C1*> ...)))

Boundary conditions.

```
(bctab
     (<bc-name>
          (basephase liquid)
          (range "<elem-range>" "<elem-range>" ...)
          (tables (C <var-table>))
          (factor  (comp <factor-table>))
     )
     ...
     (<bc-name>
          (basephase liquid)
          (range "<elem-range>" "<elem-range>" ...)
          (tables (C <var-table>))
          (factor  (comp <factor-table>))
     )
)
```

specifies boundary condition information for the transport model.

Component and phase properties.

```
(compprop (comp (liquid
                  (halfLife <Thalf>)
                  (freeDiffusivity <coef>)
                )
        )
)
```

set component dependent properties for transport model.

<Thalf> *real*

half-life of contaminant (units must be in seconds) in the liquid phase if the statement `(decay on)` is present (see subsection on `decay`). If, in addition, the statement `(decay-on-solid on)` is present, the half-life also refers to decay of contaminant that is adsorbed the solid; otherwise, there is no decay on the solid.

<coef> *real*

free diffusion coefficient in liquid phase (m^2/s).

```
(phaseprop (liquid (tortuosity <tort-fun>)))
```

sets phase dependent properties for transport model.

<tort-fun> *word*

name of the correlation for the tortuosity factor τ which multiplies the free liquid diffusion coefficient. Currently, the only legal value is `Millington` which specifies the Millington correlation: $\tau = \phi^{1/3} S_\ell^{7/3}$ where ϕ is porosity and S_ℓ is liquid saturation. (The diffusion flux is given by $q = -S_\ell \phi \tau D \nabla C$ where D is the free diffusion coefficient.) If a real number is placed instead of a word, then that number is used for the tortuosity factor.

Contaminant decay.

`(decay on)`

(optional) if present, then contaminant decays in the liquid phase with half-life equal to the value set by `halfLife` in `compprop`. If not present of if this statement is replaced by `(decay off)`, then there is no decay of contaminant.

`(decay-on-solid on)`

(optional) if present, then contaminant adsorbed onto the solid phase will decay with half-life equal to the value set by `halfLife` in `compprop`. The option `decay` for decay in the liquid phase must be also be on in order to have decay on the solid. If the `decay-on-solid` is not present of if this statement is replaced by `(decay-on-solid off)`, then there is no decay of contaminant adsorbed on the solid.

7. References

Lee, K., A. Kulshrestha, and J. Nitao, Interim Report on Verification and Benchmark Testing of the NUFT Computer Code, Lawrence Livermore National Laboratory Report UCRL-ID-113521, 1993.

Nitao, J.J., Reference manual for the NUFT flow and transport code, Lawrence Livermore National Laboratory Report UCRL-ID-113520, 1995.

A. The Mathematical Model

We derive the balance equations. There are two components, water (w) and contaminant (c), in a single liquid phase (ℓ). The gas phase is assumed to be at a constant, spatially uniform pressure conditions. The contaminant in the water phase is non-volatile. The solid phase is assumed nondeformable.

The balance equations for water and contaminant components in the liquid phase are given by

$$\frac{\partial \phi \rho_\ell S_\ell \omega_\ell^\gamma}{\partial t} = -\nabla \cdot \phi \rho_\ell S_\ell (\omega_\ell^\gamma \mathbf{V}_\ell + \mathbf{J}_{h\ell}^\gamma + \mathbf{J}_\ell^{*\gamma_\ell}), \qquad \gamma = w, c \tag{1}$$

where Fickian laws for dispersive and diffusive fluxes are given by

$$\mathbf{J}_{h\ell}^\gamma = -\boldsymbol{D}_{h\ell}^\gamma \nabla \omega_\ell^\gamma \tag{2}$$

$$\mathbf{J}_\ell^{*\gamma} = -D_\ell^{*\gamma} \nabla \omega_\ell^\gamma \tag{3}$$

and Darcy's law gives

$$S_\ell \phi \mathbf{V}_\ell = -\frac{\boldsymbol{k}_\ell(S_\ell)}{\mu_\ell} \cdot (\nabla p_\ell + \rho_\ell g \nabla z) \tag{4}$$

The capillary pressure relationship when the flow is unsaturated is given by

$$p_\ell = -p_c(S_\ell) \tag{5}$$

where we take the pressure datum is chosen such that the ambient gas pressure is zero pressure. The various variables are defined as

ω_ℓ^γ	mass fraction of γ-component in phase ℓ
S_ℓ	liquid saturation
ϕ	porosity
ρ_ℓ	liquid density
$\mathbf{V}_\ell$	liquid phase velocity
$\mathbf{J}_{h\ell}^\gamma$	hydrodynamic dispersive flux
$\mathbf{J}_\ell^{*\gamma}$	molecular diffusive flux
$\boldsymbol{D}_{h\ell}$	dispersion tensor
$D_\ell^{*\gamma}$	diffusion coefficient
$\boldsymbol{k}_\ell$	permeability function
μ_ℓ	liquid phase viscosity
p_ℓ	liquid phase pressure
p_c	capillary pressure function

In addition we have the constraint

$$\omega_\ell^w + \omega_\ell^c = 1 \tag{6}$$

Summing the two balance equations gives the balance equation for the liquid phase

$$\frac{\partial \phi \rho_\ell S_\ell}{\partial t} = -\nabla \cdot (\phi \rho_\ell S_\ell \mathbf{V}_\ell) \tag{7}$$

We now make the assumption that the contaminant is dilute, i.e. $\omega^c \ll \omega^w$ so that all of the terms in the balance equation are not functions of ω_ℓ^c. This equation can be solved for S_ℓ or p_ℓ as the primary variable. All constitutive relationships are functions of the primary variable.

The transport of the contaminant obeys the balance equation

$$\frac{\partial \phi \rho_\ell S_\ell \omega_\ell^c}{\partial t} = -\nabla \cdot \phi \rho_\ell S_\ell (\omega_\ell^c \mathbf{V}_\ell + \mathbf{J}_{h\ell}^c + \mathbf{J}_\ell^{*c_\ell}). \tag{8}$$

Numerically, we first solve at each time step, Eq. (7). Then solve Eq. (8) using the flow velocity $\mathbf{V}_\ell$ and saturation S_ℓ calculated from Eq. (7).

A.1. Variable Switching

When, for an element, the liquid pressure is negative $p_\ell < 0$, the flow equation uses as the primary variable, S_ℓ. If the pressure reaches zero, the primary variable is switched to p_ℓ. The primary variable is switched back to saturation if the two conditions are satisfied: (1) pressure becomes negative and (2) there is a neighboring connected element that satisfied these two conditions in the previous time step.

A.2. Numerical Discretization

An implicit-in-time scheme (Euler's method) is used for the flow equation (7),

$$(U\phi\rho_\ell S_\ell)_i^{(n+1)} - (U\phi\rho_\ell S_\ell)_i^{(n)} = \Delta t^{(n)} \sum_{j\in\mathcal{N}_i} (\mathbf{q}_\ell)_{i,j}^{(n+1)} \tag{9}$$

where the phase mass flux is given by

$$(\mathbf{q}_\ell)_{i,j} = -(A\rho_\ell \boldsymbol{k}/\mu)_{i,j} \left[\left((p_\ell)_i - p_\ell)_j \right) / L_{i,j} + \rho_{i,j}\, g\beta_{i,j} \right] \tag{10}$$

$$L_{i,j} = L_{i,j}^{(1)} + L_{i,j}^{(2)} \tag{11}$$

$$\boldsymbol{k}/\mu = \frac{L_{i,j}}{(L_{i,j}^{(1)}/(\boldsymbol{k}/\mu)_i) + (L_{i,j}^{(2)}/(\boldsymbol{k}/\mu)_j)} \tag{12}$$

$$\rho_{i,j} = \frac{L_{i,j}^{(1)}\rho_i + L_{i,j}^{(2)}\rho_j}{L_{i,j}} \tag{13}$$

$$\Delta t^{(n)} = t^{(n+1)} - t^{(n)} \tag{14}$$

Superscripts refer to the time-level and subscripts to element index. Note that (9) is a non-linear equation in the saturation at time level $n+1$.

An implicit-in-time scheme is used for the transport equation,

$$(\mathcal{U}\phi\rho_\ell S_\ell \omega_\ell^c)_i^{(n+1)} - (\mathcal{U}\phi\rho_\ell S_\ell \omega_\ell^c)_i^{(n)} = \Delta t^{(n)} \sum_{j\in\mathcal{N}_i} \left[(\omega_\ell^c)^{(n+1)} (\mathbf{q}_\ell)_{i,j}^{(n+1)} + (\mathbf{q}_\ell^{\text{diff}})_{i,j}^{(n+1)} \right] \tag{15}$$

where the diffusive flux is given by

$$(\mathbf{q}_\ell^{\text{diff}})_{i,j} = -\left(\frac{A\rho_\ell D}{L} \right)_{i,j} (\omega_i - \omega_j) \tag{16}$$

$$D_{i,j} = \frac{L_{i,j}}{(L_{i,j}^{(1)}/D_i) + (L_{i,j}^{(2)}/D_j)} \tag{17}$$

An option for an explicit-in-time scheme is also available.

B. Sample Problems

B.1. Flow Problem

Description:

This sample problem is a two-dimensional infiltration problem in a cylindrically symmetric system with "point" constant head source. The model has eight layers with the top layer simulating the boundary condition at the top of the model. Elements at the topmost layer are all inactive except for the innermost element which is the source kept at constant head. The source is kept at a constant head of 3.5 meters for 200 minutes and then turned off afterwards. The bottommost layer is kept at constant zero head and saturated conditions to simulate the aquifer. The lateral boundary is kept at no flow conditions.

Input file:

```
;;;file: us1-sam1.in;;;
;; sample problem

(us1p  ;; name of flow model

     (title "pt21")           ;; run title
     (tstop 23.1h)            ;; stopping time is 23.1 hours
     (stepmax 1000)           ;; maximum no. of time steps
     (time 0.0)               ;; initial time
     (dtmax 1.e30)            ;; maximum time step
     (dt 1m )                 ;; initial time step is 1 minute

     ;; set output formats
     (output
          ;; outputs all liquid saturation into file "pt21.Sl"
          (field (format list) (range "*") (variables Sl)(file-ext ".Sl")
                 (outtimes 0 39m 70m 102m 222m 287m 342m 23h)
          )
          ;; output liquid phase flux from source into file "pt21.influx"
          (flux-history  (phase-flux liquid)
                         (index-con 1 1 1  1 1 2) (file-ext ".influx")
          )
          ;; force model to take small time steps when source shuts off
          (forcetimes (outtimes 200m 201m))
     ) ;; end output

     ;; set material properties
     (rocktab
          ;; sandy material
          (HI    (porosity  0.25)
                 (Kx 1.5e-5)(Ky 0.)(Kz 0.)
                 (pc pcVanGen (Sr 0.4)(m 0.6)(alpha 5.0)(Sj 0.41))
                 (kr krVanGen (Sr 0.4) (m  0.6))
          )
          ;; moderate permeability material
          (MOD   (porosity  0.3)
                 (Kx 1.5e-6)(Ky 0.)(Kz 0.)
                 (pc pcVanGen (Sr 0.4)(m 0.3)(alpha 1.0)(Sj  0.41))
```

```
            (kr krVanGen (Sr 0.4) (m  0.3))
      )
      ;; clayey material
      (LO    (porosity  0.3)
             (Kx 1.5e-7)(Ky 0.)(Kz 0.)
             (pc pcVanGen (Sr 0.4)(m 0.2)(alpha 0.5)(Sj  0.41))
             (kr krVanGen (Sr 0.4) (m  0.2))
      )
     ;; pseudo soil type for source element above surface
      (SRC   (porosity  0.99)
             (Kx 1.e-2)(Ky 0.)(Kz 0.)
             (pc 0.0)  ;; zero cap. pressure
             (kr 1.0)    ;; rel. perm. set to unity
      )

) ;; end rocktab

;; set boundary conditions
(bctab
     ;; put flux into the source element
     (src (basephase liquid)
          (range "S*")
          ;; turn source on from 0 to 200 min. and off after that
          (factor  (water  0.0 1.0  200m 1.0 201m 0.0 1.e30 0.0))
          ;; set element to constant head of 3.5 meters
          (tables
                  (Sl  0. 1.0  1.e30 1.0)
                  (H   0. 3.5  1.e30 3.5)
          )
     )

     ;; keep water table elements at saturated conditions
     (WT  (basephase liquid)
          (range "W*")
          (tables
                  (Sl  0. 1.0  1.e30 1.0)
                  (H   0. 0.0  1.e30 0.0)
          )
     )
 )  ;; end bctab

;; initial conditions
(state
     (Sl by-key ("S*" 1.0 ) ("I*" 1.0) ("H*" 0.2) ("M*" 0.3) ("L*" 0.8)
                ("W*" 1.0)
     )
     (H  by-key   ("*"    0.0))  ;; note that bctab will overwrite
                                 ;; these values by any set there
) ;; end state

;; generate mesh
(genmsh
   (coord cylind)
   ;; note that increasing z coordinate will be downward
```

```
        (down 0. 0. 1.)

        ;; set subdivisions in radial r direction
        (dx
               0.05  ;; 0.05 radius source
               0.1   ;; r = 0.15
               0.2   ;; r = 0.35
               0.4   ;; r = 0.75
               0.5   ;; r = 1.25
               0.5
               0.5
               0.5   ;; note: nr = 8
        )

        ;; set angle subdivision
        (dy 360.)

        ;; set subdivisions in z direction
        (dz
               0.01  ;; set first row of elements which are inactive
                     ;; except for i=1 for surface flux
                1.0   ;; sandy material layer
                1.0   ;; moderate material layer
                1.0   ;; sandy material layer
                1.0   ;; clay material layer
                1.0   ;; sandy material layer
                0.01  ;; water table k=7
        )

        ;; set material type and element name prefix
        (mat
                (S SRC   1 1  1 1  1 1)    ;; source element
                (I NULL 2 8  1 1  1 1)    ;; inactive elements
                (H HI    1 8  1 1  2 2)    ;; sandy layer
                (M MOD   1 8  1 1  3 3)    ;; moderate perm
                (H HI    1 8  1 1  4 4)    ;; sandy
                (L LO    1 8  1 1  5 5)    ;; clay
                (H HI    1 8  1 1  6 6)    ;; sandy
                (W HI    1 8  1 1  7 7)   ;; water table elements
        )
    ) ;; end genmsh

) ;;; end of input
```

B.2. Transport Problem

Description:

same two-dimensional infiltration problem as above but with transport of a conservative tracer in the source water.

Input file:

```
;;;file: us1sam2.in;;;
;; sample problem

;; beginning of common data
(common
     (title "pt21")         ;; run title
     (tstop 23.1h)            ;; stopping time is 23.1 hours
     (time 0.0)               ;; initial time
     (genmsh
        (coord cylind)
        ;; note that increasing z coordinate will be downward
        (down 0. 0. 1.)

        ;; set subdivisions in radial r direction
        (dx
               0.05  ;; 0.05 radius source
               0.1   ;; r = 0.15
               0.2   ;; r = 0.35
               0.4   ;; r = 0.75
               0.5   ;; r = 1.25
               0.5
               0.5
               0.5   ;; note: nr = 8
        )
        ;; set angle subdivision
        (dy 360.)
        ;; set subdivisions in z direction
        (dz
               0.01  ;; set first row of elements which are inactive
                     ;; except for i=1 for surface flux
                1.0   ;; sandy material layer
                1.0   ;; moderate material layer
                1.0   ;; sandy material layer
                1.0   ;; clay material layer
                1.0   ;; sandy material layer
                0.01  ;; water table k=7
        )

        ;; set material type and element name prefix
        (mat
                (S SRC   1 1  1 1  1 1)    ;; source element
                (I NULL  2 8  1 1  1 1)     ;; inactive elements
                (H HI    1 8  1 1  2 2)    ;; sandy layer
                (M MOD   1 8  1 1  3 3)    ;; moderate perm
                (H HI    1 8  1 1  4 4)    ;; sandy
```

```
               (L LO    1 8  1 1  5 5)    ;; clay
               (H HI    1 8  1 1  6 6)    ;; sandy
               (W HI    1 8  1 1  7 7)    ;; water table elements
        )
     ) ;; end genmsh
) ;; end of common data

;; beginning of flow model input
(us1p                          ;; name of flow model

     (stepmax 1000)            ;; maximum no. of time steps
     (dtmax 1.e30)             ;; maximum time step
     (dt 1m )                  ;; initial time step is 1 minute

  ;; the rest of the data for the flow model is same as the above
  ;; flow-only problem

                        ......

) ;;; end of flow model input

;;; beginning of transport input data
(us1c

     (stepmax 10000)
     (dtmax 1.e30)
     (dt  3.)

     (output
          ;; output concentrations C into file "pt21.C"
          (field (format list) (file-ext ".C") (range "*") (variables C)
                (outtimes 0 39m 70m 102m 222m 287m 342m 23h)))

     (state
          (C by-key    ("*"      0.0))) ;; initial concentrations are zero

     (rocktab
          (HI  (porosity 0.25)(Kd  0.))   ;; conservative tracer
          (MOD (porosity 0.3)(Kd   0.))
          (LO  (porosity 0.3)(Kd   0.))
          (SRC (porosity 0.99)(Kd  0.))
     )

     (compprop (comp (liquid (freeDiffusivity 1.e-9))))

     (phaseprop (liquid (tortuosity Millington)))

     (bctab
        (source
            (basephase liquid)
            (factor  (comp 0.     1.0    ;; shut off solute source
                           200m   1.0    ;; after 200 m inc.
                           201m   0.0    ;; turn off diffusion flux
                         1.e30    0.0))  ;; from source element
```

```
                (range "S*")
                (tables
                        (C  0. 1.
                            1.e30  1.0)))

        (bottom
                (basephase liquid)
                (range "W*")
                (tables
                        (C  0. 0.0 1.e30 0.0)))  ;; keep WT at zero concen.
        )
)
```